

## **FINDING TECHNIQUE FOR A SCHEDULING SYSTEM**

### **CROSS-REFERENCE TO RELATED APPLICATION**

This application incorporates by reference and claims the benefit of U.S. Provisional Application No. 60/193,834, filed March 31, 2000; U.S. Provisional Application No. 60/193,917, filed March 31, 2000; U.S. Provisional Application No. 60/193,832, filed March 31, 2000; U.S. Provisional Application No. 60/193,705, filed March 31, 2000; and U.S. Provisional Application No. 60/193,833, filed March 31, 2000.

### **TECHNICAL FIELD**

The technical field relates generally to scheduling. More particularly, it pertains to finding an opening in a schedule within a predetermined period of time.

### **COPYRIGHT NOTICE - PERMISSION**

A portion of the disclosure of this patent document contains materials which are subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure as it appears in the Patent and Trademark Office patent files or records, but otherwise reserves all copyright rights whatsoever. The following notice applies to the software and data as described below and in the drawings attached hereto: Copyright © 2001, MDSI Mobile Data Solutions Inc., All Rights Reserved.

### **BACKGROUND OF THE INVENTION**

To schedule a desired service to be performed, a customer calls a service organization to place an order for the desired service. The customer may be kept on the phone for an undesirably long period of time while the service organization searches the schedule in an attempt to fit the order into the schedule. To avoid annoying the customer, some service organizations do not search the schedule while the customer is on the phone, but simply accept the order and dump it into a time bucket without scheduling the order to a particular shift of a

worker. The night before the order is to be performed, the service organization executes a batch process that assigns orders to various workers. One problem with this approach is that the service organization lacks the ability to accurately assess the capacity of the workforce to do work.

The approach as described above limits the ability of a customer to specify his preferences in regard to when and how the order is to be performed. The customer is unable to select a desired appointment window or a desired worker with a set of desired skills. As the expectations of customers have increased over time, scheduling systems that fail to meet increased customer expectations may result in customer dissatisfaction and lead to the eventual lack of acceptance in the marketplace. Thus, what is needed are systems and methods for enhancing the scheduling process that allow customers' increased expectations to be met.

## SUMMARY OF THE INVENTION

An illustrative aspect of the invention includes a method for scheduling. The method includes forming a list of shifts in a schedule and determining a shift in which an order can be scheduled. The method also includes assigning the order to the shift that is determined by the act of determining. The acts of forming, determining, and assigning are contemporaneously executed while a customer is on a phone attempting to schedule the order.

Another illustrative aspect of the invention includes a method for finding an opening to fit an order in a schedule. The method includes computing an amount of free time required in a shift to fit the order and creating a schedulable time block from a virtual free time block. The schedulable time block includes a primary block, at least one expansion block, and at least one load block. The method also includes examining the primary block. Under the act of examining, the primary block is a candidate to fit the order if the duration of the primary block, excluding at least one break, is greater than or equal to the amount of free time required in the shift to fit the order. The method further includes creating at least one opening in the shift from the schedulable time block so as to present a customer with at least one option of fitting the order in the schedule to perform a desired service.

Another illustrative aspect includes a method for assigning an order to an opening in a schedule after a customer has selected an appointment window in the schedule. The opening and the appointment window are specified. The method includes generating a list of schedulable time blocks for a shift identified in the opening and then intersecting the opening and the appointment window to obtain a time range. The method also includes choosing the opening to assign the order if a schedulable time block from the list of schedulable time blocks includes the opening. The opening is within the time range obtained by the act of intersecting.

Another illustrative aspect includes a method for assigning an order to a schedule after a customer has specified an appointment window in the schedule. The method includes checking a list of openings for overlap with the appointment window and generating a list of schedulable time blocks in a shift if there is no overlap. The method includes assigning the order to the schedule if there is an opening in the list of openings that overlaps with the appointment window or an opening in the list of schedulable time blocks that overlaps with the appointment window.

#### BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of a system showing the relationship between a service organization, a scheduling system, mobile service representatives, and customers according to one aspect of the present invention.

Figure 2 is a block diagram of a scheduling system according to one aspect of the present invention.

Figure 3 is a process diagram of a method for finding an opening in a schedule to fit an order to a shift of a worker and assigning the order to the opening according to one aspect of the present invention.

Figure 4 is a process diagram of a method for finding an opening in a schedule to fit an order to a shift of a worker according to one aspect of the present invention.

Figure 5 is a process diagram of a method for computing an amount of free time required to accommodate an order in a shift.

Figures 6A-6B are a process diagram of a method for examining a primary block according to one aspect of the present invention.

Figure 7 is a process diagram of a method for computing extra time that can be obtained by relocating assigned orders earlier in the shift or later in the shift according to one aspect of the present invention.

Figures 8A-C are graphical diagrams showing graphs to illustrate techniques for relocating a portion of a shift to earlier in time or later in time to find extra time to fit an order.

#### DETAILED DESCRIPTION OF THE INVENTION

In the following detailed description of exemplary embodiments of the invention, reference is made to the accompanying drawings, which form a part hereof, and in which are shown, by way of illustration, specific exemplary embodiments in which the invention may be practiced. In the drawings, like numerals describe substantially similar components throughout the several views. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention. Other embodiments may be utilized and structural, logical, electrical, and other changes may be made without departing from the spirit or scope of the present invention. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present invention is defined only by the appended claims. The present application incorporates by reference the following reference: Guy Druce, Level-1 Algorithm, v. 1.6 (December 4, 2000).

Figure 1 is a block diagram of a system 100 according to one aspect of the present invention. The system 100 includes a service organization 102. The service organization 102 performs services for a roster of customers 108. When a customer 108 is interested in having a service performed by the service organization 102, the customer 108 calls the service organization 102 to make a reservation for the service performed.

Using the scheduling system 104, the service organization 102 negotiates with the customer 108 to place a reservation in a schedule. The scheduling system 104 provides to the service organization 102 several appointment windows from which the customer 108 may

choose. The customer 108 selects an appointment window in which the service requested will be performed.

During the negotiation, the scheduling system 104 determines which mobile service representatives 106 can do the work. The customer 108 picks an appointment window and the scheduling system 104 assigns to the order to a mobile service representative. A mobile service representative 106 is one who is capable of performing the service requested by the customer 108. Each time a reservation is placed, the scheduling system 104 accounts for many factors in assigning the reservation to a mobile service representative 106, such as time availability, skill sets, geographic area, duration of each job, travel times between jobs, and equipment requirements. As customers 108 place more reservations for services, the scheduling system 104 periodically optimizes the schedule by relocating reservations assigned to a single mobile service representative or swapping reservations between different mobile service representatives.

The scheduling system 104 can be implemented over one machine or several machines with different processors. This allows the scheduling system 104 to be scalable depending on the needs of the service organization 102. The scheduling system 104 is also configurable. This allows the scheduling system 104 to meet different constraints and objectives. In one embodiment, the scheduling system 104 is implemented over Common Object Request Broker Architecture (CORBA). In another embodiment, the scheduling system 104 is implemented using a database, such as an Oracle database.

Figure 2 is a block diagram of a scheduling system 200 according to one aspect of the present invention. The scheduling system 200 is similar to the scheduling system 104 of Figure 1. The scheduling system 200 includes a finder 202. When a customer calls the service organization to place an order for service, the finder 202 finds one or more openings in the schedule in which the order may fit. Recall that prior systems may take a prohibitively long period of time to find openings in the schedule. In contrast, the embodiments of the present invention through the implementation of the finder 202 return one or more openings within an amount of time so that the customer need not wait on the phone for an undesirably long period of time.

After the finder 202 finds one or more openings, the negotiator 204 may present these openings as appointment windows to the service organization. The service organization can discuss these appointment windows with the customer and negotiate with the customer to select an appointment window to perform the service in accordance with the order. The customer may make specific demands, such as a request for a particular worker to perform the service or a request for a particular set of skills in the performance of the service. The finder 202 can find openings in the schedule associated with a particular worker.

When the customer has selected an appointment window, the assigner 206 proceeds to assign the order to the opening associated with the appointment window. When the order is assigned, the optimizer 208 optimizes the schedule so as to accommodate additional orders. In one embodiment, the optimizer 208 defragments the available spaces in a shift to fit additional orders in the shift.

Figure 3 is a process diagram of a method 300 for finding an opening in a schedule to fit an order to a shift of a worker according to one aspect of the present invention. The method 300 begins with an act 302 for generating a list of shifts. The act 302 generates the list of shifts using a window over the schedule. The window helps to control the amount of information to find each shift in the list of shifts. Each shift in the list of shifts may contain enough time to accommodate the order. The act 302 also includes a set of candidacy rules that allow the act to be more selective about each shift in the list of shifts.

The method 300 includes an act 304 for generating a list of virtual free time blocks. The list of virtual free time blocks is generated for a shift of a worker. A virtual free time block is an amount of time that could be used to fit an order by either bumping one or more contiguous orders within a shift or by a free time block. The term “free time block” means a block of free time between the end of the work on one order and the start of travel to another order. Thus, by bumping one or more contiguous orders within a shift, a virtual free time block may accumulate enough free time to fit an order into the shift. The term “contiguous” means next to or near in time or sequence.

The method 300 includes an act 306 for iterating through each shift in the list of shifts generated in the act 302. The act 306 allows a set of acts to execute for each shift in the

list of shifts. This set of acts help to find one or more openings in the list of shifts so as to fit an order into the schedule for a customer. Recall that the list of shifts is generated from a window over the schedule to control the amount of information that must be processed to find one or more openings. This set of acts is described in greater detail hereinbelow.

The method 300 includes an act 308 for iterating through each virtual free time block from a list of virtual free time blocks. The list of virtual free time blocks is generated from the act 304 for each shift in the list of shifts generated in the act 302. In one embodiment, each virtual free time block is a free time block. Another abstraction of the free time block called a schedulable time block keeps track of the virtual free time blocks that can be gathered to fit an order into a shift. The schedulable time block is discussed in greater detail hereinbelow.

The act 308 allows a set of acts to execute for each virtual free time block in the list of virtual free time blocks. This set of acts help to identify whether the virtual free time block is a candidate to become an opening in the shift so as to fit an order. The combination of act 306 and act 308 searches through the list of shifts and the virtual free time blocks in the list of shifts for one or more openings to fit an order.

The method 300 includes an act 310 for finding an opening. The act 310 comprises other acts that are executed to find one or more openings to fit an order. The act 310 is iterated by the act 308 for each virtual free time block in the list of virtual free time blocks and by the act 306 for each shift in the list of shifts.

The method 300 also includes an act 312 for assigning an order. When one or more openings are found, these openings are presented to a customer. The customer selects a desired opening. The act 312 attempts to assign the order to the opening. If the order is successfully assigned to the opening, the order is assigned to a shift of a worker. On the appointed date and time, the worker proceeds to perform the service as specified in the order.

Figure 4 is a process diagram of a method 400 for finding an opening in a schedule to fit an order to a shift of a worker according to one aspect of the present invention. The method 400 discusses the act 310 of Figure 3 in greater detail. The method 400 includes an act 402 for computing an amount of free time required to accommodate the order. The amount of free time required to accommodate the order is a sum of three terms. One of the three terms is

a travel time, another is a job time, and the third term is a difference travel time. These terms are explained by proceeding to Figure 5.

Figure 5 is a process diagram of a method 500 for computing an amount of free time required to accommodate an order in a shift. The method 500 includes an act 502 for calculating a travel time  $T$  between a first activity and a second activity. The travel time  $T$  is defined as the time to travel from a location  $a$  to another location  $b$ . The travel time  $T$  can be mathematically described as  $T = t(a,b)$ . The term “ $t$ ” is a function that returns a time period for traveling between the location  $a$  and the location  $b$ .

The method 500 also includes an act 504 for calculating a difference travel time. The difference travel time can be explained by an example. Suppose the shift comprises  $sabFcd$ . The term “ $s$ ” signifies the start of the shift. The terms “ $a$ ,” “ $b$ ,” “ $c$ ,” and “ $d$ ” are work activities that are to be performed by a worker. The term “ $F$ ” signifies a virtual free time block. Thus, the time travel to  $a$  equals  $t(s,a)$ . The time travel to  $b$  equals  $t(a,b)$ . The time travel to  $F$  equals 0. The time travel to  $c$  equals  $t(b,c)$ . The time travel to  $d$  equals  $t(c,d)$ .

Suppose that a new order  $n$  is to be inserted into the virtual free time block  $F$ . The time travel to  $n$  equals  $t(b,n)$ . And the difference travel time is mathematically defined as:  $\delta = t(n,c) - t(b,c)$ . Thus, the difference travel time is a result of the travel time from the new order to the activity  $c$  subtracting the travel time from the activity  $b$  to the activity  $c$ . This difference travel time is the time calculated by the act 504.

The method 500 includes an act 506 for calculating a job time. The job time is the amount of time required to perform a service as specified by an order. The method 500 sums the travel time  $T$  between a first activity and a second activity, the difference travel time, and the job time to form a summation. This summation is the amount of free time required in a shift to fit the order.

Returning to Figure 4, act 402 produces the amount of free time required in a shift from a sum of the travel time  $T$  between a first activity and a second activity, the difference travel time, and the job time. The method 400 also includes an act 404 for creating a schedulable time block from a virtual free time block. The schedulable time block includes a primary block. The schedulable time block may include at least one expansion block, at least one load block, or both.

The primary block is a block of time into which the order may be inserted. The primary block may not have adequate time to fit an order however. The expansion block and the load block help to provide some extra time that may increase the time of the primary block to fit the order. The expansion block finds extra time in the shift by relocating other assigned orders in the shift. The load block finds extra time in the shift by removing assigned orders in the shift so that the total load of the shift is kept below a worker's load limit.

The method 400 includes an act 406 for examining the primary block to see if the order can be fitted into the primary block. The act 406 comprises a set of acts that determine whether primary block is a candidate to fit the order. This set of acts may try to find extra time in the shift by shuffling assigned orders earlier or later in time in the shift or by removing assigned orders. To focus on this set of acts in greater detail, the discussion proceeds to Figures 6A-6B.

Figures 6A-6B are a process diagram of a method 600 for examining a primary block according to one aspect of the present invention. The method 600 includes an act 602 for examining the primary block. The act 602 deems that the primary block is a candidate to fit the order if the duration of the primary block, excluding any breaks, is greater than or equal to the amount of free time required in the shift to fit the order. An act 604 follows the act 602 to check whether the primary block is a candidate. If the primary block is a candidate, the method 600 progresses to node 613B to other acts.

If the primary block is not a candidate, this indicates that the primary block has insufficient time to fit the order. To find extra time, the method 600 progresses to an act 606 for finding the extra time by relocating a portion of the shift that is filled with assigned orders to the right of the primary block. In other words, the relocation of the portion of the shift is a shuffle of assigned orders to later in the time of the shift so as to create extra time that can be added to the primary block to fit an unassigned order. If extra time is found, the extra time is added as an expansion block to the schedulable time block created in the act 404 of Figure 4. In order to focus in greater detail on how the extra time in the shift is found, the discussion further proceeds to Figure 7.

Figure 7 is a process diagram of a method 700 for computing extra time that can be obtained by relocating assigned orders earlier in the shift or later in the shift according to one

aspect of the present invention. The method 700 begins at the act 702 for calculating an amount of time available in a virtual free time block for fitting an order. This amount of time is represented as  $t$ .

The method 700 includes an act 704 for computing an amount of time in which a portion of the shift must be relocated. This amount of time is represented as  $Dt$ .  $Dt$  equals  $EOT - t$ .  $EOT$  is the amount of free time required in a shift to fit the order, which is calculated by act 404 of Figure 4.  $t$  is the amount of time available in the virtual free time block for fitting an order as discussed above.

When  $Dt$  is computed, the method 700 progresses to an act 706 for comparing  $Dt$  to zero. If  $Dt$  is a negative quantity, this indicates that there is enough time in the virtual free time block to fit the order, and the method 700 ends its execution. If  $Dt$  is a positive quantity, this indicates that a portion of the shift must be relocated, and the method 700 progresses to the act 708.

The act 708 computes an amount of time in which the portion of the shift can be relocated. This amount is represented as  $t_{min}$ . The portion of the shift can be relocated earlier in time or later in time relative to the virtual free time block in which an order is inserted. By relocating the portion of the shift, more free time may become available to add to the virtual free time block to accommodate the order.

Visually, relocating the portion of the shift to the left means relocating the portion of the shift to an earlier point in time. Correspondingly, relocating the portion of the shift to the right means relocating to a later point in time. To illustrate in greater detail the computation of the amount of time in which the portion of the shift can be relocated, the discussion proceeds to Figures 8A-C.

Figures 8A-C are graphical diagrams showing graphs to illustrate techniques for relocating a portion of the shift to earlier in time or later in time to find extra time to fit an order. Figure 8A is a graphical diagram showing a graph 800 to illustrate a technique for relocating a portion of the shift. The graph 800 includes a shift 802 that defines the work period of a worker.

The shift 802 includes a virtual free time block 804. For the purpose of this illustration, the virtual free time block 804 is the time block into which an order is to be fitted.

The shift 802 includes other assigned orders, such as order 806, order 808, and order 810. Order 806 is designated as Order 1. Order 808 is designated as Order 2. And order 810 is designated as Order 3. The shift 802 includes travel time to get from one assigned order to the next. The shift 802 includes appointment windows 812, 814, and 816. The appointment window 812 defines a time period during the shift 802 in which a worker could start to work on Order 1. The appointment window 814 similarly defines a time period in which the worker could start to work on Order 2. And the appointment window 816 defines a time period for a worker to start to work on Order 3.

The graph 800 also illustrates a maximum relocation for each of the Orders 1, 2, and 3. There are three maximum relocation figures, 818, 820, and 822. The maximum relocation figure denotes the maximum relocation that can be done for a particular assigned order. To relocate to a point later in time, the maximum relocation is formed from taking the absolute value of the subtraction of the starting time of an assigned order from the end of the appointment window for the assigned order. To relocate to a point earlier in time, the maximum relocation is formed from taking the absolute value of the subtraction of the starting time of an assigned order from the start of the appointment window for the assigned order. To calculate  $t_{\min}$ , which is the amount of time the portion of the shift can be relocated, the minimum of the three maximum relocation figures 818, 820, and 822 is taken. Thus, in the graph 800, it can be seen that the maximum relocation figure 822 is the minimum.

To summarize what has been discussed hereinbefore, suppose a new order is to be inserted in place of the virtual free time block 804. Because the virtual free time block 804 is insufficient to accommodate the new order, an extra amount of time is required. This amount of time is  $Dt$ . The extra amount of time can be obtained by relocating Orders 1, 2, and 3. For example, Orders 1, 2, and 3 can be relocated to the right of the virtual free time block 804. Order 2 has the least amount of time available between the end of its appointment window and the start time of the Order 2. Hence, the maximum amount of time in which the Orders 1, 2, and 3 may be relocated is given by  $t_{\min}$ .  $t_{\min}$  is given by the maximum relocation figure 822 as shown in the graph 800.

Figure 8B is a graphical diagram illustrating a technique for relocating a portion of the shift to a later point in time. The graph 800 includes a shift 802<sub>1</sub>. The shift 802<sub>1</sub> is similar to the shift 802 discussed in Figure 8A but is different in that the shift 802<sub>1</sub> includes another virtual free time block 824 in addition to the virtual free time block 802. Recall that the virtual free time block 804 is the block in which the order is to be inserted. Given the presence of the virtual free time block 824, Orders 1, 2, and 3 and the travel time associated with these orders can be relocated to the right. The maximum amount of relocation is given by  $t_{min}$ . Thus, in the shift 802<sub>2</sub>, the virtual free time block 804 has been enlarged by  $t_{min}$ , and correspondingly, the virtual free time block 824 has been compressed by  $t_{min}$ .

Figure 8C is a graphical diagram illustrating a technique for relocating a portion of the shift to an earlier point in time. The graph 800 of Figure 8C is similar to the graph 800 of Figure 8B except for the location of the virtual free time blocks 804 and 824. Recall that the virtual free time block 804 is the block in which the order is to be inserted. Given the presence of the virtual free time block 824, Orders 1, 2, and 3 and the travel time associated with these orders can be relocated to the left. The maximum amount of relocation is given by  $t_{min}$ . Thus, in the shift 802<sub>2</sub>, the virtual free time block 804 has been enlarged by  $t_{min}$ , and correspondingly, the virtual free time block 824 has been compressed by  $t_{min}$ .

Returning to act 708 of Figure 7,  $t_{min}$  is now computed. Thus, at this point in the method 700, two quantities are known: the amount of time by which the portion of the shift must be relocated  $Dt$  and the amount of time by which the portion of the shift can be relocated  $t_{min}$ . The method 700 progresses to an act 712 for comparing  $t_{min}$  to  $Dt$ .

If  $t_{min}$  is less than  $Dt$ , this means that no extra time can be found in the shift even with the relocation of the portion of the shift, and the method 700 progresses to an act 710. The act 710 loops to the next virtual free time block in the list of virtual free time blocks as discussed in the act 308 of Figure 3. Otherwise, if  $t_{min}$  is greater than  $Dt$ , this means that there is extra time to fit the order, and the method 700 completes its execution.

Returning to the act 606 of Figure 6A, recall that the act 606 finds the extra time by relocating the portion of the shift to the right of the primary block. If the relocation succeeds in finding enough time so that the extra time added to the duration of the primary block is greater

than or equal to the amount of free time required in the shift to fit the order, then the primary block can be used to fit the order. In this case, the extra time is added as expansion block to the schedulable time block created in the act 404 of Figure 4.

An act 608 follows the act 606 to check whether the primary block is a candidate. If the primary block is a candidate, the method 600 progresses to the node 613B and to other acts. Otherwise, the method 600 progresses to an act 610.

The act 610 finds the extra time by relocating the portion of the shift to the left of the primary block. In other words, the method 600 attempts to relocate the portion of the shift to an earlier point in time so as to find extra time in which to fit the order. If the relocation succeeds in finding enough time so that the extra time added to the duration of the primary block is greater than or equal to the amount of free time required in the shift to fit the order, then the primary block can be used to fit the order. The extra time is added as expansion block to the schedulable time block created in the act 404 of Figure 4.

An act 612 follows the act 610 to check whether the primary block is a candidate. If the primary block is a candidate, the method 600 progresses to the node 613B and to an act 616 of Figure 6B. Otherwise, the method 600 progresses to a node 613A and to an act 614. The act 614 eliminates the virtual free time block from further consideration. The virtual free time block is derived from the schedulable free time block that was created in the act 404 of Figure 4. The next virtual free time block is then considered as discussed in act 308 of Figure 3.

If the method 600 progresses to the act 616, the virtual free time block from which the schedulable time block is derived can be expanded to suit the order. However, because each shift has a load limit, the addition of the amount of free time required to fit the order to the shift may exceed this load limit. If the load limit is not exceeded, the method 600 completes its execution.

Otherwise, the method 600 progresses to an act 618 to reduce the load of the shift so as to fit the order. In order to reduce the load of the shift, the act 618 finds at least one virtual free time block in the shift whose orders can be removed to reduce the load. If the act 618 finds an order in a virtual free time block to be removed, that virtual free time block will be added to

the load block, which was created in the act 404 of Figure 4. The method 600 then terminates and returns to the end of the act 406 of Figure 4.

If the act 618 cannot find any virtual free time block in the shift that can be removed, the method 600 progresses to an act 620. The act 620 eliminates the virtual free time block, from which the schedulable time block was derived, from further consideration, and the method 600 terminates its execution. The next virtual free time block is then considered as discussed in the act 308 of Figure 3.

Returning to the end of the act 406 of Figure 4, recall that the act 406 examines the primary block to see whether the order can be fit into the primary block. The act 406 comprises a set of acts that determine whether the primary block is a candidate to fit the order. This set of acts, which are discussed in greater detail in Figures 6A-6B, may try to find extra time in the shift by relocating assigned orders earlier or later in time in the shift or by removing assigned orders. The method 400 progresses to an act 408.

The act 408 creates one or more openings from the schedulable time block created in the act 404. These openings are added to a list of openings that are returned to the service organization. The service organization may use these openings to define a range of times in which work on the order may start. This can be used to negotiate with a customer to set a time for an appointment to perform the order.

## CONCLUSION

A technique has been discussed for finding one or more openings in a schedule so as to allow a service organization to address the preferences of customers while allowing the service organization to satisfy constraints placed on the scheduling system and to meet business objectives. A scheduling system comprises three main components. The negotiator interacts with customers to negotiate an appointment window in which the service is performed as specified in the reservation. The assigner assigns the reservation to a shift of a desired mobile service representative. And the optimizer continuously optimizes the schedule in the background. The embodiments of the present invention focus on providing one or more openings within a predetermined period of time for the negotiator to interact with customer in a

timely manner. Because customers dislike waiting on the phone for an extended period of time, the embodiments of the present invention enable the negotiator to work fast enough to obtain an agreeable time window and worker to perform the service requested.

Although the specific embodiments have been illustrated and described herein, it will be appreciated by those of ordinary skill in the art that any arrangement, which is calculated to achieve the same purpose, may be substituted for the specific embodiments shown. This application is intended to cover any adaptations or variations of the present invention. It is to be understood that the above description is intended to be illustrative, and not restrictive. Combinations of the above embodiments and other embodiments will be apparent to those of skill in the art upon reviewing the above description. The scope of the invention includes any other applications in which the above structures and fabrication methods are used. Accordingly, the scope of the invention should only be determined with reference to the appended claims, along with the full scope of equivalents to which such claims are entitled.